

Capacitación formadores Generación de capacidades en el Ecosistema Digital de Bogotá

Formación especializada TI

Control de versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo

Tipos

Sistemas de control de versiones locales

Sistemas de control de versiones centralizados

Sistemas de control de versiones distribuidos

GIT

Instalando y configurando GIT

Tu identidad

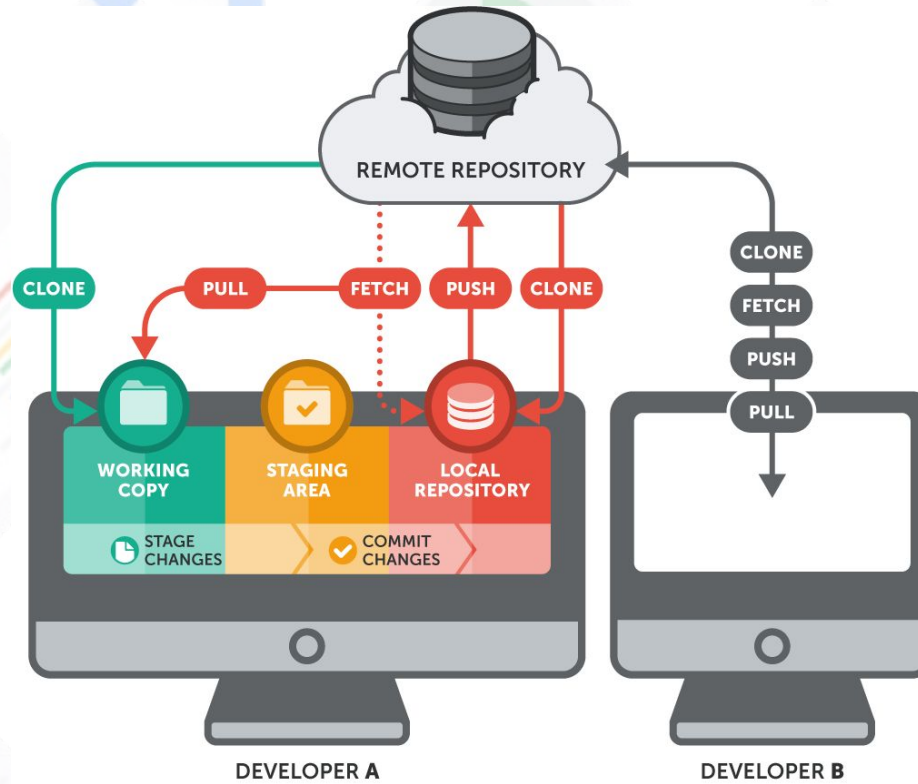
```
$ git config --global user.name "John Doe"  
$ git config --global user.email johndoe@example.com
```

Comprobando tu configuración

```
$ git config --list
```


Crear cuenta en Github y un repositorio de prueba

Git



Crea un repositorio nuevo

Crea un directorio nuevo, ábrelo y ejecuta

`git init`

para crear un nuevo repositorio de git.

Clonando un repositorio existente

```
$ git clone [url]
```

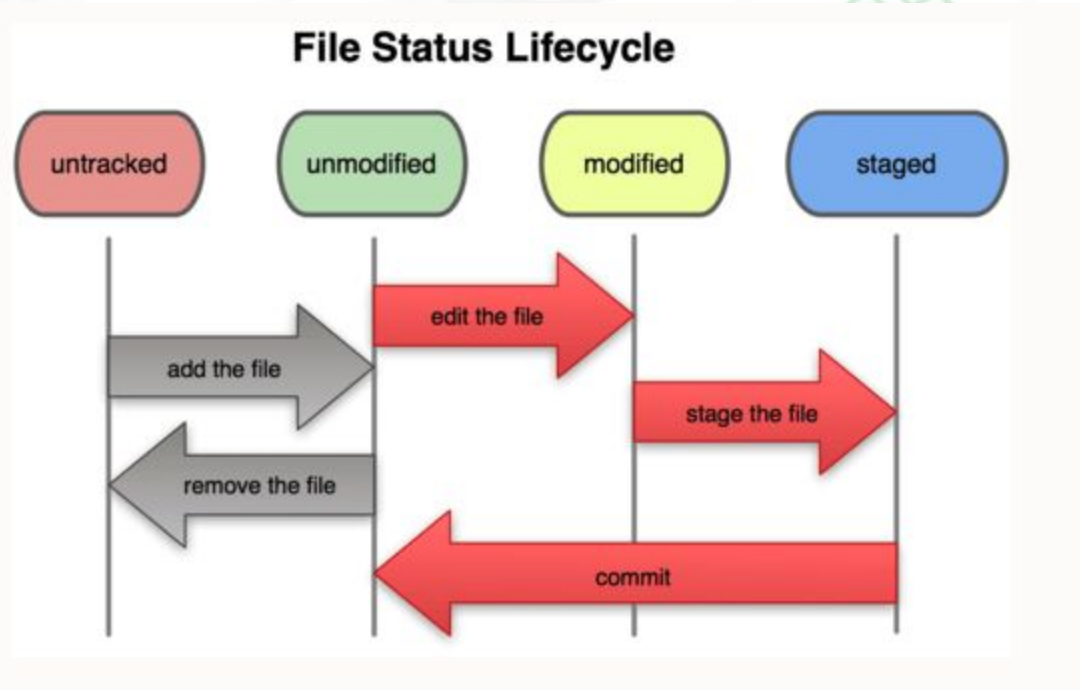
```
$ git clone git://github.com/schacon/grit.git
```

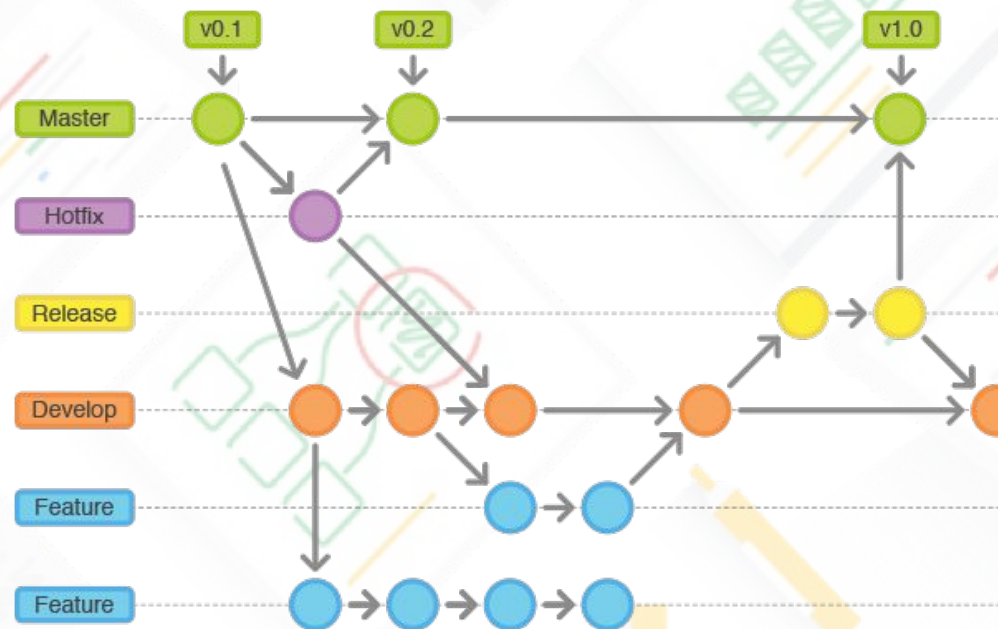
Fundamentos

a continuación se recomienda explicar cada uno de los estados de los archivos para comprender como funciona GIT



Guardando cambios - Estados de los archivos





Comprobando estado de archivos

```
$ git status
```

Agregando archivos

`$ git add [file]` (agrega solo un archivo)

O

`$ git add .` (agrega todos los cambios en el repo)

Confirmar cambios cambios

```
$ git commit -m 'initial commit'
```

Estado del repositorio

\$ git log

creemos un archivo .gitignore y hagamos un ejemplo de ignorar archivos de repo en el que se esté trabajando

Ignorando archivos

.gitignore

ignore doc/notes.txt, but not doc/server/arch.txt

doc/*.txt

ignore all .txt files in the doc/ directory

doc/**/*.*

Repositorios Remotos

```
$ git clone git://github.com/schacon/ticgit.git
```

```
$ cd ticgit
```

```
$ git remote -v
```

Añadiendo repositorios remotos

```
$ git remote
```

```
origin
```

```
$ git remote add pb git://github.com/paulboone/ticgit.git
```

```
$ git remote -v
```

```
origin git://github.com/schacon/ticgit.git
```

```
pb git://github.com/paulboone/ticgit.git
```

Enviando cambios a Github

```
$ git push -u origin rama
```

Actualizando repositorio local

\$ git pull

O

git fetch

git merge [remote]/[branch]

Trabajando con Ramas

Una rama Git es simplemente un apuntador móvil apuntando a un Commit. La rama por defecto de Git es la rama master. Con la primera confirmación de cambios que realicemos, se creará esta rama principal master apuntando a dicha confirmación

Crear una branch

\$ git branch testing

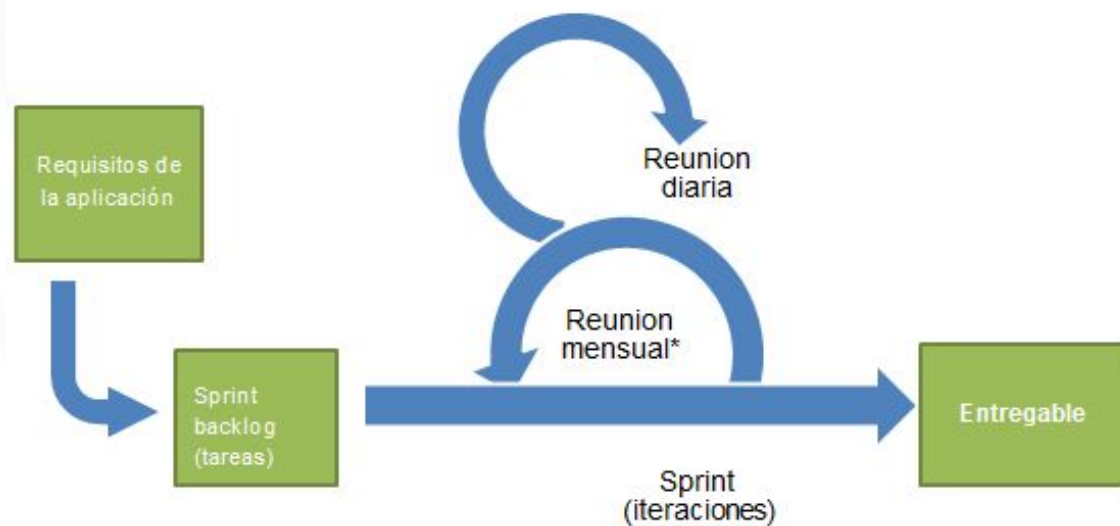
\$ git checkout testing

Pull Request

Haremos un *pull request* cuando queramos contribuir con nuestros cambios -mejoras, corrección de errores, actualizaciones- a un repositorio que ya existe.



Piensa Digital
<Tecnología real para soluciones reales/>



Taller

<https://try.github.io>